The "Script execute" module
PRINTED MANUAL

©2006-2008 AGG Software

"Script execute" module

©2006-2008 AGG Software

All rights reserved. No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of information contained in this document or from the use of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: 17.09.2008

Publisher

AGG Software

Production

©2006-2008 AGG Software
http://www.aggsoft.com

Table of Contents

Part 1 Introduction 1

Part 2 System requirements 1

Part 3 Installing Script execute 2

Part 4 Glossary 5

Part 5 User Manual 6

1 Configuration 6

2 Supported functions 7

Missing features 7

Data types 8

Mathematical 8

String functions 9

Date and Time 9

Miscellaneous functions 9

Special functions 10

Part 6 Troubles? 11

1 Possible problems 11

©2006-2008 AGG Software

1 Introduction

The filter module "Script execute" for our data loggers (for example, Advanced Serial Data Logger or Advanced TCP/IP Data Logger) is an interpreter for scripts written in most popular programming languages. Unique feature of this plug-in is ability to use several languages (PascalScript, C++Script, JScript and BasicScript), so you can write scripts using your favorite language. The plug-in doesn't use Microsoft Scripting Host, so it can be used on any computer with Windows.

"Script execute" combines cross-platform scripting, fast code execution, small footprint, rich set of features and a splendid scalability. Make your applications the most flexible and powerful ones with "Script execute".

It is a smart tool easy in use. Supports most standard functions and operators for all languages. Within scripts you can change values of parser variables or/and add new variables to the export. The plug-in allows you to filter data packets and send data to a port.

- Multi-language architecture allows you to use a number of languages (at present moment PascalScript, C++Script, BasicScript, JScript).
- Standard language set: variables, constants, procedures, functions (nested functions allowed) with var/const/default parameters, all the standard operators and statements (including case, try/finally/except, with), types (int, float, bool, char, string, multi-dimensional array, enum, variant), classes (with methods, events, properties, indices and default properties).
- Types compatibility checking.
- Access to any some standard class (for example, TStrings, TFileStream).
- Allows storing values between executions.
- Allows sending string, bytes or data arrays to a port.
- Can be used with multiple ports and in multiple configurations.

2 System requirements

The following requirements must be met for "Script execute" to be installed:

Operating system:
- Windows 95;
- Windows 95 OSR2;
- Windows 98;
- Windows Me;
- Windows NT4;
- Windows 2000;
- Windows XP (all editions) (both x86 and x64);
- Windows 2003 Server (both x86 and x64);
- Windows Vista (both x86 and x64).

Free disk space: Not less than 5 MB of free disk space is recommended.
Special access requirements: You should log on as a user with Administrator rights in order to install this module.

The main application (core) must be installed, for example, Advanced Serial Data Logger.

©2006-2008 AGG Software

Notes for Microsoft Vista:

Since our software saves data to the registry and installs to the Program Files folder, the following requirements must be met:

1. You need Administrator rights to run and install our software
2. The shortcut icon of our software will be located on the desktop;
3. Windows Vista will ask for your confirmation to continue the installation.

NOTE: You can configure the user account only once in order not to see the above dialog box any more. Search Google for the solution of this problem.

3 Installing Script execute

1. Close the main application (for example, Advanced Serial Data Logger) if it is running;
2. Copy the program to your hard drive;
3. Run the module installation file with a double click on the file name in Windows Explorer;
4. Follow the instructions of the installation software. Usually, it is enough just to click the "Next" button several times;
5. Start the main application. The name of the module will appear on the "Modules" tab of the "Settings" window if it is successfully installed.

If the module is compatible with the program, its name and version will be displayed in the module list. You can see examples of installed modules on fig.1-2. Some types of modules require additional configuration. To do it, just select a module from the list and click the "Setup" button next to the list. The configuration of the module is described below.

You can see some types of modules on the "Log file" tab. To configure such a module, you should select it from the "File type" list and click the "Advanced" button.

©2006-2008 AGG Software

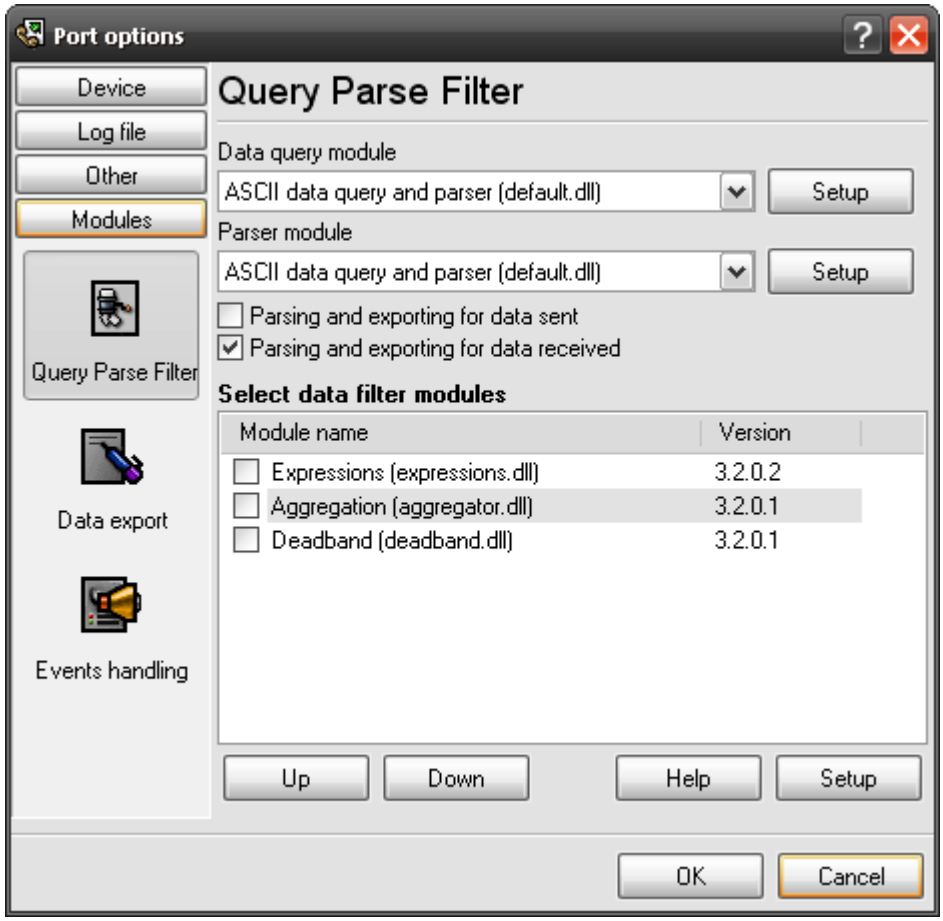


Fig.1. Examples of installed modules

©2006-2008 AGG Software

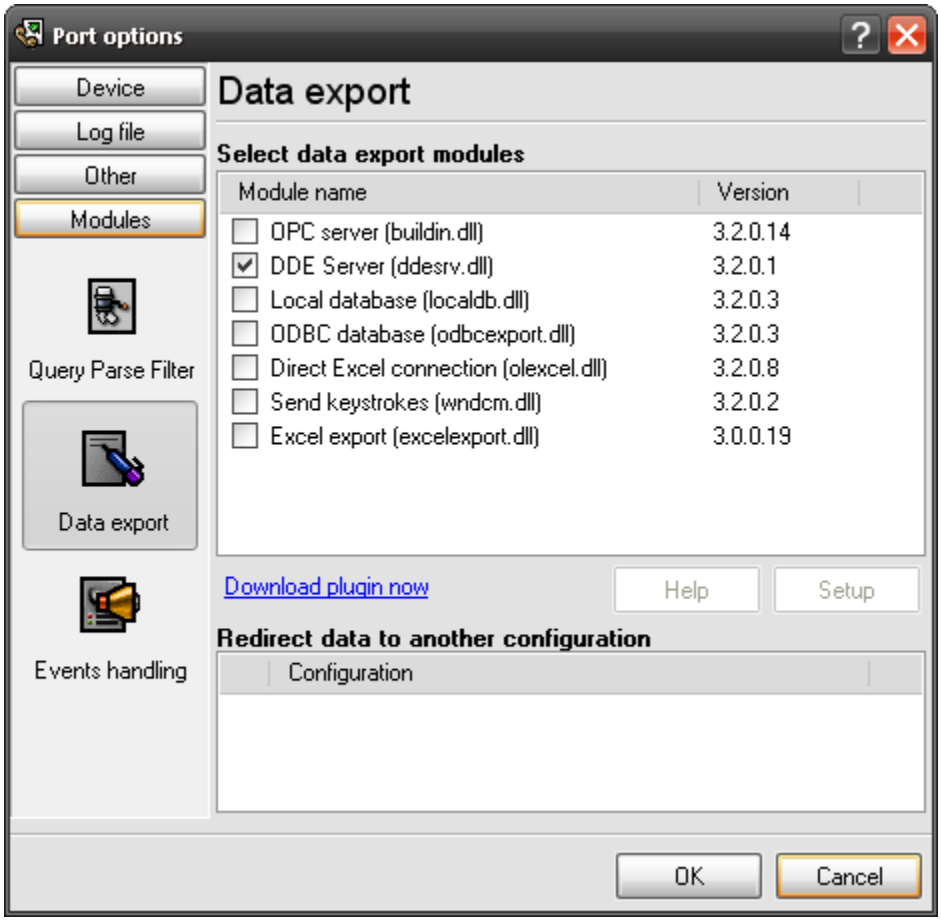


Fig.2. Examples of installed modules

©2006-2008 AGG Software

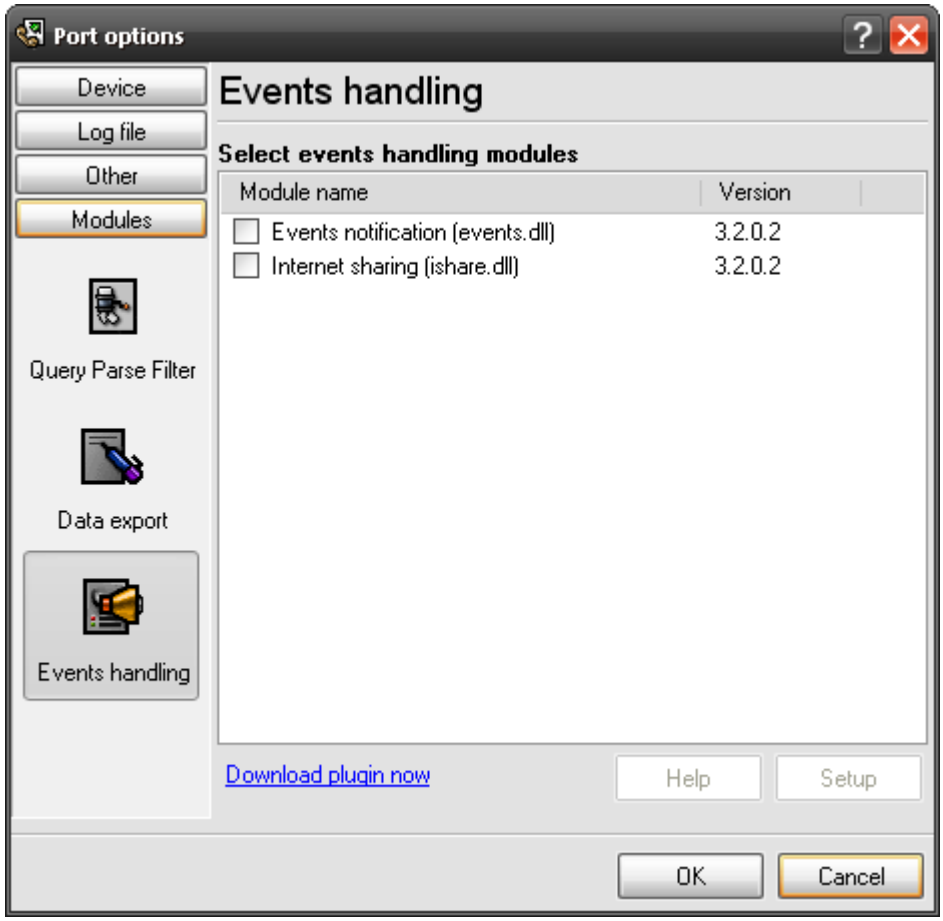


Fig.3. Examples of installed modules

4 Glossary

Plug-in - module

Main program – the program shell that uses this module. For example: Advanced Serial Data Logger

Parser – the module that processes the data flow singling out data packets from it and variables from data packets. These variables are used in data export modules after that.

Core - see "Main program".

©2006-2008 AGG Software

5 User Manual

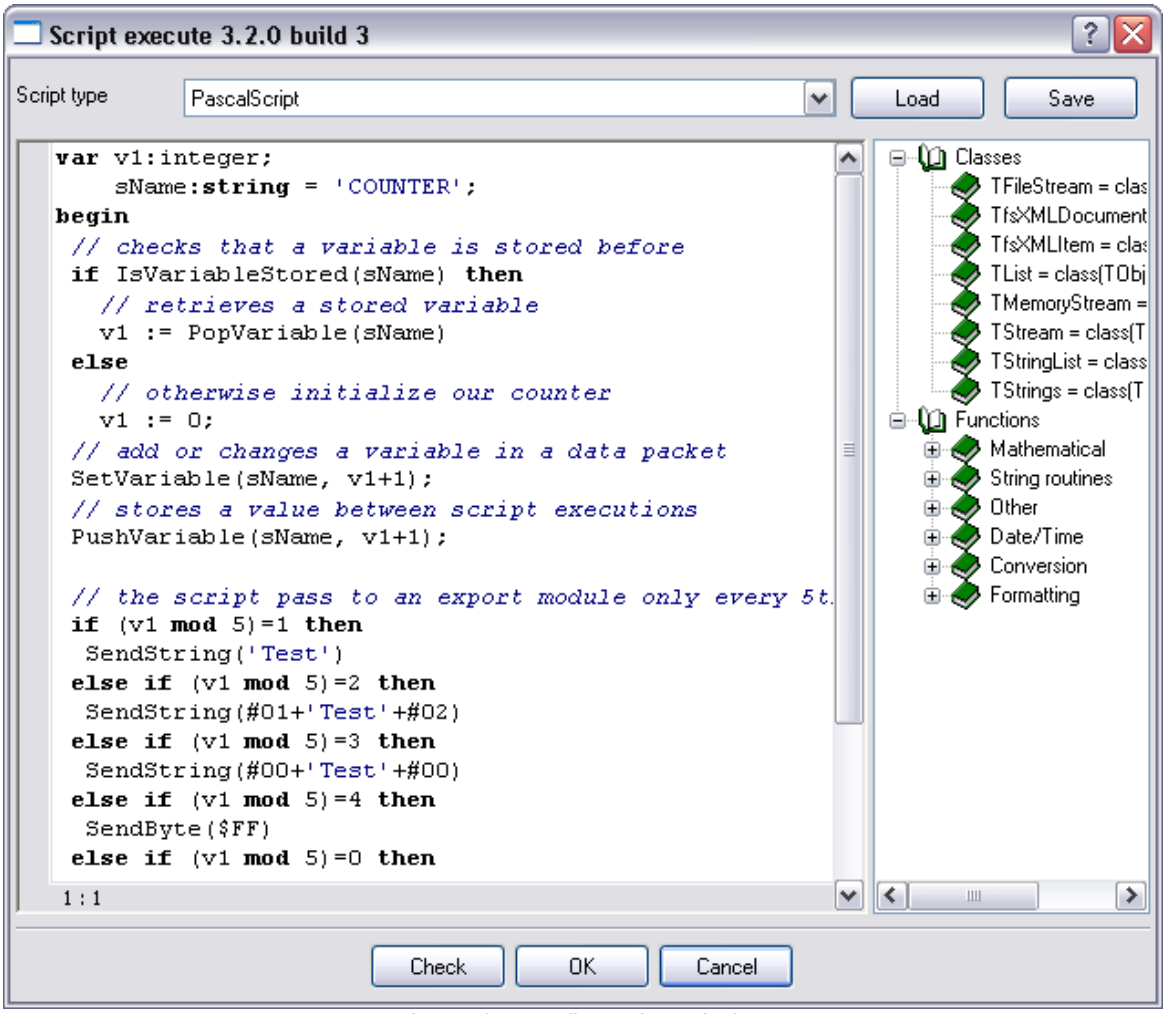
5.1 Configuration

The module configuration is very simple (pic.1). You may select a script type and specify a script in the editor window. You can add classes or functions from the left tree by double-click.

All supported functions are described in following sections.

When you'll complete editing, you should click the "Check" button and verify your script. Then you can click the "OK" button and the plug-in will apply all your changes.

You can load and save your scripts to a file by clicking "Load" or "Save" buttons.



Pic.1. The configuration window

Hot keys

Key	Action

©2006-2008 AGG Software

Cursor arrow	Cursor moving
PgUp, PgDn,	Page Up / Page Down
Ctrl+PgUp	Move to the begin of text
Ctrl+PgDn	Move to the end of text
Home	Move to the begin of line
End	Move to the end of line
Enter	Move to the next line
Delete	Delete symbol at right or selected text
Backspace	Delete symbol at left
Ctrl+Y	Delete current line
Ctrl+Z	Undo last change
Shift+Стрелки курсора	Select the text block
Ctrl+A	Select all text
Ctrl+U	Unindent selected block
Ctrl+I	Indent selected block
Ctrl+C, Ctrl+Insert	Copy to clipboard
Ctrl+V, Shift+Insert	Paste from clipboard
Ctrl+X, Shift+Delete	Cut to clipboard
Ctrl+Shift+<цифра>	Set bookmark
Ctrl+<цифра>	Goto bookmark
Ctrl+F	Search text
F3	Continue search

5.2 Supported functions

5.2.1 Missing features

- No type declarations (records, classes) in the script; no records, no pointers, no sets (but you can use 'IN' operator - "a in ['a'..'c','d']"), no shortstrings, no GOTO statement.
- C++Script: no octal constants; no 'break' in the SWITCH operator (SWITCH works like Pascal CASE); '++' and '--' operators are possible only after the variables, that is '++i' is not allowed; '--', '++' and '=' operators do not return a value, that is 'if(i++)' is not allowed; all the identifiers are case-insensitive;
- NULL constant is the Pascal Null - use nil instead of NULL.

©2006-2008 AGG Software

5.2.2 Data types

Internally the plug-in operates with the Variant type and is based on it. Nevertheless, you can use the following predetermined types in your scripts:

```
Byte      | Same as Integer type
Word      |
Integer   |
Longint   |
Cardinal  |

Boolean   | Boolean type

Real      | Same as Extended type
Single    |
Double    |
Extended  |
TDate     |
TTime     |
TDateTime |

Char      | Char type

String    | String type

Variant   | Same as Variant type
Pointer   |

Array     | Array type
```

C++Script maps some types to standard types:

int, long = Integer
void = Integer
bool = Boolean
float = Extended

JScript has no types, all types are variants. BasicScript may have types (for example, dim i as Integer), or may have no types and even no variable declaration. In this case a variable will have Variant type.

Not all of these types can be assign-compatible. Like in Object Pascal, you can't assign Extended or String to an Integer. Only one type - the Variant - can be assigned to all the types and can get value from any type.

5.2.3 Mathematical

```
Abs(e: Extended): Extended
ArcTan(X: Extended): Extended
Cos(e: Extended): Extended
Exp(X: Extended): Extended
Frac(X: Extended): Extended
Int(e: Extended): Integer
Ln(X: Extended): Extended
Pi: Extended
Round(e: Extended): Integer
Sin(e: Extended): Extended
Sqrt(e: Extended): Extended
Tan(X: Extended): Extended
Trunc(e: Extended): Integer
```

©2006-2008 AGG Software

5.2.4 String functions

```
Chr(i: Integer): Char
CompareText(s, s1: String): Integer
Copy(s: String; from, count: Integer): String
Delete(var s: String; from, count: Integer)
Insert(s: String; var s2: String; pos: Integer)
Length(s: Variant): Integer
Lowercase(s: String): String
NameCase(s: String): String
Ord(ch: Char): Integer
Pos(substr, s: String): Integer
SetLength(var S: Variant; L: Integer)
Trim(s: String): String
Uppercase(s: String): String
```

5.2.5 Date and Time

```
Date: TDateTime
DayOfWeek(aDate: TDateTime): Integer
DaysInMonth(nYear, nMonth: Integer): Integer
DecodeDate(Date: TDateTime; var Year, Month, Day: Word)
DecodeTime(Time: TDateTime; var Hour, Min, Sec, MSec: Word)
EncodeDate(Year, Month, Day: Word): TDateTime
EncodeTime(Hour, Min, Sec, MSec: Word): TDateTime
IsLeapYear(Year: Word): Boolean
Now: TDateTime
Time: TDateTime
```

5.2.6 Miscellaneous functions

Other

```
CreateOleObject(ClassName: String): Variant
Dec(var i: Integer; decr: Integer = 1)
Inc(var i: Integer; incr: Integer = 1)
InputBox(ACaption, APrompt, ADefault: string): string
InputQuery(ACaption, APrompt: string; var Value: string): Boolean
MessageDlg(Msg: string; DlgType: TMsgDlgType; Buttons: TMsgDlgButtons; HelpCtx: Longint): Integer
RaiseException(Param: String)
Random: Extended
Randomize
ShowMessage(Msg: Variant)
ValidDate(cDate: String): Boolean
ValidFloat(cFlt: String): Boolean
ValidInt(cInt: String): Boolean
VarArrayCreate(Bounds: Array; Typ: Integer): Variant
VarType(V: Variant): Integer
```

Conversion

```
DateTimeToStr(e: Extended): String
DateToStr(e: Extended): String
FloatToStr(e: Extended): String
IntToStr(i: Integer): String
StrToDate(s: String): Extended
StrToDateTime(s: String): Extended
StrToFloat(s: String): Extended
StrToInt(s: String): Integer
StrToTime(s: String): Extended
TimeToStr(e: Extended): String
VarToStr(v: Variant): String
```

©2006-2008 AGG Software

Formatting

```
Format(Fmt: String; Args: array): String
FormatDateTime(Fmt: String; DateTime: TDateTime): String
FormatFloat(Fmt: String; Value: Extended): String
FormatMaskText(EditMask: string; Value: string): string
```

5.2.7 Special functions

```
procedure SetVariable(Name: String; Value: Variant)
```

The procedure changes or adds a variable in a data packet. Later this variable can be used in a data export plug-in.

```
function GetVariable(Name: String):Variant
```

The procedure retrieves a value of a variable in a data packet. If the variable doesn't exist in the data packet, then the function returns Null.

```
procedure PushVariable(Name: String; Value: Variant)
```

The procedure stores a named value in an internal storage. Later you can retrieve this value and use in your script. If the value exists in the storage, then the procedure will overwrite it. The internal storage is empty on starting and will be cleared if the module configuration is changed.

```
function PopVariable(Name: String):Variant
```

The procedure retrieves a named value, previously stored by PushVariable. If the variable had not been stored before, then the function returns Null.

```
procedure DeleteVariable(Name: String)
```

The procedure deletes a variable in a data packet.

```
procedure DiscardDataPacket
```

Call this procedure if you don't want to export a current data packet. The module will not pass this data packet down to a data export module.

```
function IsVariableDefined(Name: String):boolean
```

With help of this function you can verify that a variable exists in a data packet.

```
function IsVariableStored(Name: String):boolean
```

With help of this function you can verify that a named value has been stored before.

```
function SendData(Data: String):boolean
```

This function allows you to send a data array defined as a string to a port. For example Data = #01+'Test'+#02

```
function SendString(Data: String):boolean
```

This function allows you to send a string value. Before sending, the data logger will replace all characters like #0D with a characters with a corresponding hexadecimal code.

```
function SendByte(Data: Byte):boolean
```

This function allows you to send a single byte.

©2006-2008 AGG Software

6 Troubles?

6.1 Possible problems

No data for publication/exporting – no data is passed for exporting. Solution: configure the parser, make sure that one or more variables are declared in the parser.

Error on binding variable with name %s [%s] – the error usually occurs if data does not correspond to the specified format. For example, the date and time format does not correspond to the data.

Unable to disconnect from the database [%s] and **Unable to connect to a database [%s]** – it is impossible to connect/disconnect to/form the database. You should check the parameters of the database connection. The analysis of the additional information will help you locate the error.

Database access error [%s]. Stop operations with the database? – the message appears if an error occurs during an attempt to execute an SQL query if the second variant of reacting to errors is selected. The message implies a "Yes" or "No" answer. The analysis of the additional information will help you locate the error.

Unable to verify your SQL script [%s] – the message appears when an attempt to analyze your SQL query fails. Check if the syntax of your SQL query is correct.

Tested successfully – the message appears if your database connection is successfully tested. It requires no additional actions.

Database isn't used – the message appears if the module is temporarily disabled (the "Temporarily disabled" check box is selected) or the database name field is empty. Check the connection parameters.

Database isn't selected - the message appears if the database type is not selected. Check the connection parameters.

Database: %s – %s contains the database name. The message appears if the database connection is successful. Usually, you see it when you call the module for the first time. It requires no additional actions.

Invalid data block length (columns=%d,length=%d) – an internal application error. It means that the data sent by the parser is in an invalid format. Perhaps, you are using the module incompatible with the version of the Advanced Serial Data Logger kernel. Update the versions of both the kernel and the module.

The time of connection is not due yet (%d,%d) – the message appears during an attempt to connect to the database after the connection to it has been lost and the "Reconnect after" option is enabled. No additional actions are required.

Invalid procedure call. Bad arguments –an attempt to call the module using invalid parameters. Perhaps, you are using the module incompatible with the version of the Advanced Serial Data Logger kernel. Update the versions of both the kernel and the module.

©2006-2008 AGG Software

Writing to the database is complete - the message appears if your queue of SQL queries is successfully executed. It requires no additional actions.

Writing to the database is complete with errors – the message appears if the executing your queue of SQL queries was interrupted by an error. It requires no additional actions.

Your SQL is empty. Please, specify some SQL text first – the message appears if you do not enter the text for your SQL query. Check if the options on the "SQL queue" tab are configured correctly.

Invalid temporary path – the path to the temporary file specified by you does not exist. Enter a new path in the "Temporary folder" field on the "Errors handling" tab.

%s, %d – will be replaced by additional information.

©2006-2008 AGG Software